

Changepoint Detection in Social Networks: an Extension of the Relational Event Model

RESEARCH REPORT

Hsuan Lee (9252568)

Supervisors:

Dr. Mahdi Shafiee Kamalabad
&
Dr. Javier Garcia Bernardo

Programme: MSBBSS

Department of Methodology & Statistics

Utrecht University
the Netherlands

Date: 08.05.2023

Candidate Journal: Social Networks

FETC Case Number: 22-1870; 22-1871

Abstract

Time-stamped relationships dominate our lives, shaping the dynamic nature of our daily interactions with others. From scheduled meetings to social media interactions, the ebb and flow of our daily routines are dictated by time and the people around us. When studying social networks, the Relational Event Model (REM) is a mature model that explains the dynamic nature of these networks. However, it does not inform us when the changepoint of the social network occurs. Understanding the changepoint of a social network can be important, as it allows us to identify the key moments when the network structure underwent significant changes, which can help us better understand the evolution of the network and the underlying social dynamics at play. In this study, we propose a changepoint detection approach under the Relational Event Model (REM) structure. We employ a moving window to segment the event history into windows and fit the REM on each window. We then utilize three state-of-the-art changepoint detection algorithms: Bayesian Online Changepoint Detection (BOCPD), Pruned Exact Linear Time (PELT), and Binary Segmentation (BS), to detect changepoints across windows and compare their performances. Our analysis of synthetic data indicates that BOCPD performs well in terms of precision, as most of the changepoint windows it indicates are highly convincing to be true changepoint windows. PELT is capable of revealing most of the changepoint windows in the network, but this comes at a trade-off with its high number of false positives. BS is the least effective algorithm, as it cannot reveal as many true changepoint windows as PELT and also lacks the high precision of BOCPD. We validate our approach using Apollo 13 voice loop data, which confirms its effectiveness in real-life analysis. The results from the synthetic analysis are consistent with the real-life data, as BOCPD and PELT capture all the potential changepoints that we hypothesize, while BS missing a few.

Keywords:

Social network, Relational Event Model, Moving Window approach, Changepoints detection, Binary Segmentation, Pruned Exact Linear Time, Bayesian Online Changepoint Detection Method

1. Introduction

The relational event model (REM) is a mature model for studying real-time social interactions and predicting future events in a social network. It uses factors (e.g., gender, age, interaction inertia, etc.) that shape social interactions, known as “effects,” [7] to parameterize the interaction rates between actors. The REM assumes that these effects are constant over time. However, given the dynamic nature of social networks, the strengths of these effects can change over time. Identifying changepoints in the REM can help us understand when social network dynamics change qualitatively. For example, in a classroom setting, students may form social networks based on interests and hobbies that change over time. Identifying these changepoints can help teachers and administrators better support student engagement and social connections, ultimately leading to a more positive learning environment.

Changepoints represent sudden shifts in time series data that reflect transitions occurring across conditions[37][3]. Changepoint detection is essential in many domains and has been the focus of numerous studies. For example, Rauhameri and Salminen compared the performance of changepoint detection algorithms on IMS data from a portable ion mobility spectrometer[31]. Jarušková utilized a maximum type statistics to detect changepoints in hydrological and meteorological series[22]. van den Burg and Williams evaluated and compared the performance of changepoint detection algorithms in general situations using 37 real-world time series data from diverse application domains[13]. In the social network area, Shafiee Kamalabad and Leenders suggested using Bayes Factor to infer changepoints in the REM[36]. This approach uses the support of two hypotheses from the data to prove the existence of changepoints. However, there have been few studies on changepoint detection under the REM framework apart from this study.

The purpose of this study is to propose an approach that facilitates the identification of changepoints in the REM framework. This approach seeks to address the current gap in REM’s ability to detect changes in network dynamics. To achieve this, we utilize the moving window approach developed by Mulder and Leenders [30] in combination with REM, which we refer to as MW-REM. The MW-REM mechanism involves delineating a specific duration of time (i.e., a window) that partially overlaps with the subsequent window. This window then slides over the entire event history, allowing us to observe fluctuations in the effects over time. To detect changepoints in the social network, our method utilizes the effect fluctuations along the windows as input for the changepoint detection algorithms: Binary Segmentation (BS), Pruned Exact Linear Time (PELT), and Bayesian Online Changepoint Detection (BOCPD). These algorithms were identified as top performers in a comparison conducted by van den Burg and Williams [13] in a general application.

In this study, we specifically investigate (1) the effectiveness of our proposed changepoint detection approach under the REM structure, and (2) compare the performances of the three algorithms under our proposed method. To evaluate their feasibility and performance in the context of social network scenarios within MW-REM, we employ synthetic data to calculate metrics such as Confusion Matrix, Mean Squared Error (MSE), and Mean Signed Difference (MSD). We also apply these methods to real-life data to test their external validity. Our method is applicable to any social network scenario, including communication in surgical rooms, interactions between teachers and students, as well as cooperation and competition between companies. The detection of changepoints can facilitate a better understanding of social dynamics, identify potential challenges as they arise, and inform future strategies based on data. The blueprint of our study is depicted in Figure 1.

The structure of the paper is as follows: In the next section, the methodology part, we first introduce REM and MW-REM. Next, we introduce the three changepoint detection algorithms we utilized in our study. Then, we discuss how our proposed approach works based on MW-REM in and demonstrate how we simulate synthetic data and calculate metrics for comparing the algorithms. Finally, we introduce how we model real-life data, specifically the Apollo 13 voice loop data. In section 3, the results and discussion part, we present the effectiveness of our method and comparison results between the three changepoint algorithms on synthetic

data and real-life data.

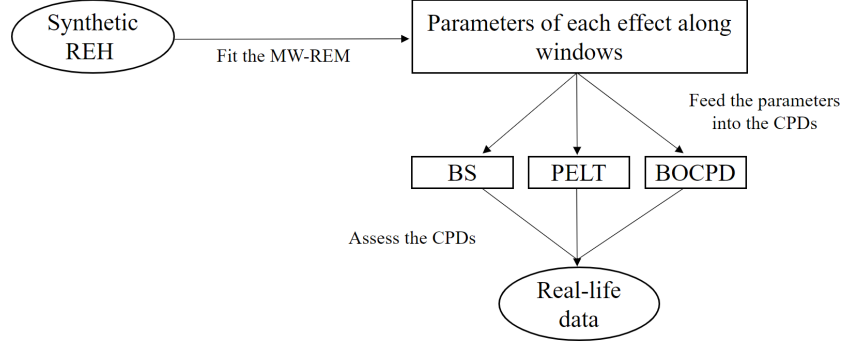


Figure 1. Flowchart Depicting the Research Design of the Present Study

2. Methodology

2.1. Relational Event Model (REM)

The REM is a powerful tool for modeling Relational Event History Data (REH), which at minimum involves sender, receiver, and time information, as shown in Table 1. By analyzing the impact of various effects on the social network, the REM enables us to understand the social interaction dynamic and forecast the timing and participants of the future events in the network. This approach parameterizes both exogenous effects, which are actor characteristics that do not depend on past interactions in the network, such as age or gender, and endogenous effects, which depend on past interactions in the network, such as transitivity or inertia.

Time (hh, mm, ss)	sender	receiver	message
13:14:05	Patel	Chen	The weather conditions look good.
13:14:09	Chen	Patel	Yes, it should be a smooth flight.
13:14:12	Nguyen	Chen	I've done my safety checks.
13:14:14	Chen	Nguyen	Great job!

Table 1. An Example of Relational Event History Data

To predict the next event in the network, the REM considers every possible sender-receiver combination (s, r) as a potential occurrence at time t , and the collection of these pairs at time t is named the risk set, denoted as $R(t)$. The risk set's size for each event is usually $N \times (N - 1)$, where N refers to the number of actors in the social network, as each actor can only be either a sender or a receiver, but not both. The REM models the event rate (λ) for each sender-receiver pair (s, r) to predict which pair will be involved in the next event and when it will occur. The event rate is assumed to remain constant between the time of the present event and the time of the following event, the pair with a higher event rate in the risk set R at time t is more likely to occur in the next event. The probability of the (s, r) pair taking place in the next event follows a multinomial distribution, given by

$$P((s, r)|t) = \frac{\lambda(s, r, t)}{\sum_{R(t)} \lambda(s, r, t)}, \quad (1)$$

where $\lambda(s, r, t)$ represents the event rate of a pair (s, r) and $R(t)$ denotes the risk set for time t .

The duration between two events follows an exponential distribution, which is given by

$$\Delta t \sim \text{Exponential} \left(\sum_{R(t)} \lambda(s, r, t) \right), \quad (2)$$

where Δt denotes the duration between two events. The higher the total event rate of the risk set, the shorter the Δt .

The event rate is typically considered as a log-linear function of the outcome in REM with specific effects, given by

$$\log \lambda(s, r, t) = \sum_p \beta_p x_p(s, r, t), \quad (3)$$

where β_p represents the parameter of effects, which expresses the strength of one effect on the entire social network, and $x_p(s, r, t)$ denotes the statistics, which can be either an exogenous or endogenous effects.

Despite the REM's outstanding capability in capturing social network dynamics and forecasting, its weakness lies in the assumption that the strength of all effects (i.e., β_p) remains constant throughout the entire event history. This assumption is unrealistic considering the dynamic property of social interactions. For instance, consider a social network where age is the exogenous effect, and transitivity (i.e., the tendency of friends to have common friends) is the endogenous effect. Initially, age may strongly influence the event rate, indicating that older individuals are more likely to interact. However, this effect may weaken or even reverse over time due to changes in the network's dynamics or cultural norms. Similarly, the strength of the transitivity effect may vary as new friendships form or old ones dissolve. Therefore, assuming that effect strengths remain constant throughout the event history can result in inaccurate predictions and an incomplete understanding of social network dynamics.

2.2. Moving Window-Relational Event Model (MW-REM)

The Moving Window approach (MW) [30] addresses the limitations of the REM. The MW involves setting up a fixed-size window, i.e., a fixed length of time, that slides over the entire Relational Event History (REH) data. Each window overlaps with the previous window, and the REM is fitted to each window (see Figure 2). We refer to this approach as MW-REM in this study. The MW-REM allows us to reveal the dynamics of each effect on the social network over time through the effect parameters (β_p) given in each window. It enables us to investigate the dynamic changes of the effect strengths in social networks over time, a feature not available in the original REM.

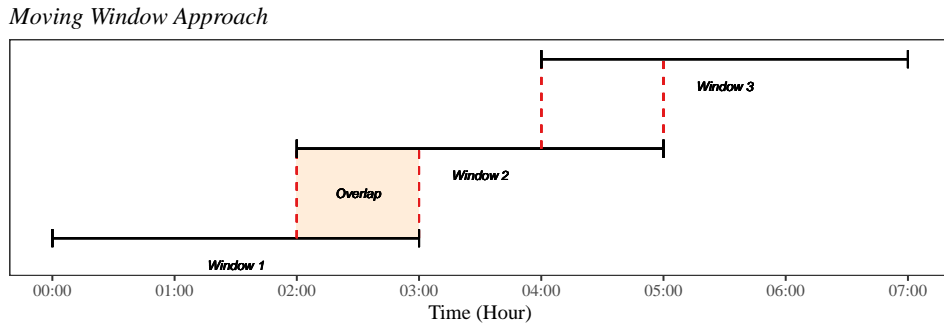


Figure 2. Example of Moving Window Approach: a window length of 1 hour with an overlap of $\frac{1}{3}$.

2.3. *Changepoint detection*

In this study, we propose an approach for detecting changepoints in REH, building on the foundation of MW-REM. As MW-REM divides the event history into partially overlapping sub-portions to fit the REM, it provides us with effect parameters (β_p) for each window over time, indicating the strength of each effect during that period, as shown in Equation 3. By fitting MW-REM to the event history, we obtain as many REMs as the fitting windows, and therefore the same number of effect parameters (β_p) as the number of fitting windows.

Our proposed approach utilizes three changepoint detection algorithms for detecting changepoints in social networks: Binary Segmentation (BS), Pruned Exact Linear Time (PELT), and Bayesian Online Changepoint Detection (BOCPD). According to van den Burg and Williams[13], these three changepoint detection algorithms were identified as top performers in a comparison in a general application. The comparison was based on criteria such as $F1$ scores and Segmentation covering metric. We chose these changepoint detection algorithms for their respective advantages and suitability for detecting changepoints in social networks, which can have abrupt changes over time. It is worth noting that changepoint detection algorithms can be classified into two categories, online and offline methods. Online methods aim to detect changes immediately in a real-time context, while offline methods examine changes retrospectively upon collection of complete data [24]. In our study, BS is an offline method, PELT and BOCPD are online methods. The fundamental mechanisms of the three changepoint detection algorithm are presented in the following sections.

2.3.1. *Binary Segmentation (BS)*

Binary Segmentation is a commonly used algorithm for detecting changepoints in time series data. The algorithm is based on a divide-and-conquer approach, which involves recursively splitting the time series into smaller segments until a segment is found that appears to be homogeneous with respect to a statistical property of interest, such as the mean or variance. Once a segment is found to be homogeneous, a changepoint is detected at the boundary between that segment and the previous one[25].

The key idea behind Binary Segmentation is to minimize a cost function that penalizes the number of changepoints and the size of the segments. The cost function can be thought of as a trade-off between model complexity and goodness of fit. A more complex model with more changepoints may fit the data better, but it will also be more likely to overfit and capture noise in the data. On the other hand, a simpler model with fewer changepoints may underfit and miss important changes in the data. The most common cost function for multiple changepoint detection is given by

$$\sum_{i=1}^{m+1} [C(y_{(\tau_{i-1}+1):\tau_i})] + \beta f(m), \quad (4)$$

where i denotes the order of a time point in a segment, m indicates the number of changepoints, τ_i implies the location of a possible changepoint (i.e., time point i). And the m changepoints will divide the data into $m + 1$ segments, with the i th segment contains $y_{(\tau_{i-1}+1):\tau_i}$. C represents the cost function of a segment, $\beta f(m)$ serves as a penalty to prevent overfitting[25].

To trade-off between accuracy and simplicity, BS starts with a single segment that spans the entire time series and recursively splits the segment into two smaller segments at the point that minimizes the cost function. This process continues until no further splits are possible, given by

$$C(y_{1:\tau}) + C(y_{(\tau+1):n}) + \beta < C(y_{1:n}), \quad (5)$$

In other words, BS searches for possible changepoints until there is no τ that meets the criteria. At this point, BS stops. Finally, the set of changepoints detected in each segment are merged to obtain the final set of changepoints.

2.3.2. Pruned Exact Linear Time (PELT)

PELT is another algorithm for detecting changepoints in time series data. Like Binary Segmentation, PELT is a divide-and-conquer approach that recursively splits the time series into smaller segments. However, PELT differs from Binary Segmentation in that it has a computational complexity that is linear with respect to the length of the time series[25], making it more efficient.

The key idea behind PELT is to iteratively remove candidate changepoints that do not significantly reduce the cost function in Equation 4. Specifically, PELT starts with a single segment that spans the entire time series, and then recursively splits the segment into smaller segments at the point that minimizes the cost function[9]. At each step, PELT evaluates the cost of adding a new changepoint between each pair of adjacent segments. If the cost of adding a new changepoint is not significant, the algorithm prunes the candidate changepoint and continues with the next pair of adjacent segments. This process is repeated until no further candidate changepoints remain.

The final set of changepoints detected by PELT is obtained by merging the remaining candidate changepoints with the ones detected in each smaller segment. The resulting set of changepoints provides an exact solution that minimizes the cost function in Equation 4 with the fewest possible number of changepoints.

2.3.3. Bayesian Online Changepoint Detection (BOCPD)

Unlike the BS and PELT, which rely on cost functions to identify changepoints, Bayesian Online Changepoint Detection (BOCPD) infers changepoints based on a Bayesian approach, which defines changepoints in terms of posterior probabilities, also known as run length probabilities, at time points.

In BOCPD, run length is an essential concept, representing the length of time elapsed since the last identified changepoint. This can be understood as akin to the segments in PELT and BS. Whenever BOCPD recognizes a changepoint, the run length drops to 0 and recalculates the length. To determine the changepoints, BOCPD calculates the run length probabilities (i.e., posterior probabilities) for each time point, which include both growth probabilities and changepoint probabilities.

According to Adams and Mackay[1], to save computational costs, it is suggested to set a cut-off point for the run length probability, typically 10^{-4} . If the run length probability reaches such a cut-off point, the time point is determined as a changepoint.

Overall, BOCPD starts by building the predictive distribution from the potential locations of changepoints, which reveals any prior knowledge regarding the data generation process. Based on the given predictive distribution, BOCPD computes the run length probability at a time point. As new data come in, the predictive distribution is continuously updated, and BOCPD iteratively runs the same procedure until no new data appear.

2.3.4. MW-REM Based Changepoint Detection

In this section, we describe the settings used for each of the three changepoint detection methods employed in our paper, as well as our proposed approach based on the MW-REM structure.

Dehling and Fried et al. have noted that the mean difference, a commonly used statistical interest in changepoint detection, is not robust to outlying observations in time series[12]. The mean difference only looks for changes in the mean value, assuming a constant variance. Similarly, variance difference looks for changes in the variance only, assuming a constant mean. To improve changepoint identification, in our study using BS and PELT, we take changes in both the mean and variance into account. In contrast, BOCPD does not rely on statistical interest such as mean or variance to identify changepoints. Instead, it uses a cut-off

point for the run length probability. In this study, we follow Adams and Mackay’s suggestion to set a cut-off point at 10^{-4} [1].

Algorithm 1 shows the details and steps of our proposed five-step approach for identifying changepoints in social networks.

Algorithm 1 MW-REM Based Changepoint Detection

- 1: **Input:** A relational event history (REH) dataset.
 - 2: Select the effects that drive social interactions using prior knowledge, statistical tests, or other methods.
 - 3: Select an appropriate window length and overlap (e.g., 2000 seconds with $\frac{2}{3}$ overlap).
 - 4: Fit the Moving Window Relational Event Model (MW-REM) to the REH dataset.
 - 5: Extract the parameters (β_p) of each effect from each window’s REM: $\log \lambda(s, r, t) = \sum_p \beta_p x_p(s, r, t)$.
 - 6: Apply Bayesian, frequentist, or other changepoint detection methods to each effect’s parameters to identify potential windows that may contain changepoints.
 - 7: **Output:** Potential changepoint windows for each effect in the social network.
-

2.4. REH data simulation

To evaluate the effectiveness of our proposed changepoint detection approach for REH data, we generated synthetic datasets with various changepoint settings. For each setting, we created 15 datasets that were simulated based on the REH characteristics. These synthetic datasets were constructed using a social network consisting of 30 actors and 10,000 events.

To construct these datasets, we randomly selected two exogenous effects and two endogenous effects. We then specified the parameter values (β_p) for each effect and its changes over time. The assignment of the effects’ parameters in the synthetic data is based on Meijerink-Bosman’s study[28] with slight adjustments. The exogenous effects include the “Sender effect,” which represents the actor’s exogenous attributes that impact their event sending rate, and the “Difference effect,” which indicates the difference in personal attributes that influence the rate of sending events. The endogenous effects consist of the “Inertia effect,” which describes the tendency of actors to repeatedly select the same receiver for their events, and the “Outdegree of the Sender effect,” which indicates the inclination of actors to send events if they have previously sent more events. The complete parameter assignments can be found in Table 2.

Using the assigned parameters of the effects, each sender-receiver pair (s, r) in the risk set R obtained the probability of occurrence at every event, as shown in Equation 1. We then built the REH by selecting the (s, r) pair with the highest probability at each event. In this study, we designed five different changepoint settings with varying numbers of changepoints. For each setting, we simulated fifteen REH datasets. The settings were categorized into three groups based on the number of changepoints they contained:

- No changepoint category:
 - All of the effects do not have any changepoints.
- One changepoint category:
 - The inertia effect has 1 changepoint, but the rest do not.
 - All of the effects have 1 changepoint.
- Two changepoints category:
 - The inertia effect has 2 changepoints, but the rest do not.
 - All of the effects have 2 changepoints.

For the setting with one changepoint, we set the changepoint for the effect(s) parameter at $t = 38200$ seconds, which corresponds to the window of 56-58. For the setting with two changepoints, we set the changepoints for the effect(s) parameters at $t = 16750$ seconds and $t = 53900$ seconds, corresponding to the windows of 24-26, and 79-81, respectively.

	<i>Sender</i>	<i>Difference</i>	<i>Inertia</i>	<i>OutdegreeSender</i>
(1) <i>All have no changepoint</i>	0.1	-0.1	0.18	0.12
(2) <i>Inertia has one changepoint</i>	0.12	-0.1	0.15 → 0.06	0.12
(3) <i>All have one changepoint</i>	0.18 → 0.04	-0.05 → -0.15	0.13 → 0.06	0.13 → 0.07
(4) <i>Inertia has two changepoints</i>	0.12	-0.11	0.01 → 0.12 → -0.01	0.12
(5) <i>All have two changepoints</i>	0.2 → -0.01 → 0.19	-0.01 → -0.12 → 0.03	0.02 → 0.14 → 0.03	-0.01 → 0.14 → 0.01

Table 2. The assignment of parameter values for the effects in the synthetic data. The rows represent the different changepoint settings, the columns represent the individual effects. The arrows indicate the parameter values before and after the changepoint.

2.5. Evaluation strategy

Having generated synthetic REH data with different changepoint settings, we proceeded to evaluate the effectiveness of our proposed changepoint detection approach on these datasets. To this end, we fitted the MW-REM with a window length of 2000 seconds and $\frac{2}{3}$ overlap to each dataset and extracted the effects' parameters. In Meijerink-Bosman's 2022 study, a window length of 2000 seconds and $\frac{2}{3}$ overlap were found to be sufficient for capturing the effect dynamics with reasonable accuracy [29]. Although this setting may miss some of the finer network details, it is computationally efficient compared to using smaller window sizes, and the effects fluctuations over time are relatively less noisy.

In the following section, we describe how we inspected the performance and compared three changepoint detection algorithms applied to these extracted parameters, utilizing three metrics: (1) the confusion matrix, (2) mean squared error (MSE), and (3) mean signed difference (MSD). Our analysis focuses on evaluating the algorithms' performance across different categories of changepoint numbers, which include no changepoint, one changepoint, and two changepoints. The classification of the synthetic data is shown in section 2.4.

2.5.1. Confusion matrix

After feeding the changepoint detection algorithms with the effects' parameters, we used the confusion matrix to evaluate the performance of each algorithm for each effect. A true positive indicates that the algorithm correctly detected a window containing the changepoint of the effect. Notably, due to the overlapping property of the windows, a changepoint can be contained in three consecutive windows simultaneously. Therefore, if an algorithm detects the changepoint in any of the three windows, it is considered a true positive. However, since the synthetic datasets are produced based on selecting the (s, r) pair with the highest rate probability in the risk set R for every event, the locations of the changepoints may differ slightly from our assignment. As a result, if a changepoint detection algorithm detects a changepoint within the range of three windows of the windows containing the true changepoint, we consider it a true positive.

A false positive indicates that the algorithm falsely detected a window as a changepoint window (i.e., a window containing a changepoint), while a false negative indicates that the

algorithm failed to detect a window containing the changepoint of the effect. Since our synthetic datasets are highly imbalanced, with many windows but few containing changepoints, we did not consider true negatives. Table 3 presents the confusion matrix for changepoint detection, showing the actual and predicted changepoints and non-changepoints.

	<i>Actual Changepoint</i>	<i>Actual None-changepoint</i>
<i>Predicted Changepoint</i>	True Positive	False Positive
<i>Predicted None-changepoint</i>	False Negative	

Table 3. Confusion Matrix for changepoint detection

Given the information from the confusion matrix, we employ three indicators to assess the effectiveness of our proposed approach and the performance of the three changepoint detection algorithms in detecting changepoints in REH. The first indicator is the number of false positive cases. We separately sum the number of false positives of the effects in no changepoint, one changepoint, and two changepoints categories for each changepoint detection algorithm. For instance, for the setting of the REH with no changepoint for all effects, the number of false positives of each effect in this setting by a changepoint detection algorithm is summed in the no changepoint category. This is done to determine which of the three changepoint algorithms has the highest likelihood of falsely detecting a changepoint when there is none, considering the no changepoint, one changepoint, and two changepoints situations in the REH.

The second indicator is the recall. Similar to the false positives indicator, we determine the performance of the changepoint detection algorithms based on the changepoint categories. However, since there are no true positives for the effects with no changepoint, we only consider the categories with one and two changepoints for each changepoint detection algorithm. The recall is calculated as

$$Recall_g = \frac{TP_g}{TP_g + FN_g}, \quad (6)$$

where TP indicates the number of true positive cases, FN indicates the number of false negative cases, and g indicates the changepoint category. Through the recall, we obtain information about the probability that each changepoint detection algorithm correctly predicts the true positive for the effects with one or two changepoints, respectively, among all positive observations.

The third indicator is the precision. Similar to recall, we assess the performance of the changepoint detection algorithms on precision in the one changepoint and two changepoint categories. The precision is calculated as

$$Precision_g = \frac{TP_g}{TP_g + FP_g}, \quad (7)$$

where TP represents the number of true positive cases, FP represents the number of false positive cases, and g represents the changepoint category. The precision indicates the probability that a predicted changepoint is indeed a changepoint, providing insight into the accuracy of the changepoint detection algorithms.

2.5.2. Mean Squared Error (MSE) & Mean Signed Difference (MSD)

We use mean squared error (MSE) and mean signed difference (MSD) as measures to examine the accuracy of the predicted changepoint window and the tendency of the algorithm to detect changepoints early or late.

The MSE measures the average squared distance between the predicted and actual changepoint windows for each changepoint detection algorithm. We compute the MSE only for true positive cases from the confusion matrix of each effect. Specifically, we calculate the MSE as[3]

$$MSE = \frac{\sum_{i=1}^{\#CP} (Predicted(CP) - Actual(CP))^2}{\#CP}, \quad (8)$$

where CP denotes the windows containing a changepoint, and $\#CP$ represents the number of changepoints in the REH. A lower MSE indicates that the algorithm's predicted changepoint window is closer to the true one in general. For the MSE evaluation, we assess the performance of the changepoint detection algorithms in the one-changepoint and two-changepoint categories. We report the average MSE of each category for each algorithm using the formula

$$Avg.MSE_g = \frac{\sum_{i=1}^{N_g} MSE_i}{N_g}, \quad (9)$$

where $Avg.MSE_g$ represents the average MSE for changepoint category g , MSE_i is the MSE value for the i th effect in changepoint category g , and N_g is the number of effects in changepoint category g .

The MSD, on the other hand, is used to examine the tendency of the changepoint detection algorithm to detect changepoints early or late. We compute the MSD only for true positive cases from the confusion matrix of each effect. Specifically, we calculate the MSD as

$$MSD = \frac{\sum_{i=1}^{\#CP} (Predicted(CP) - Actual(CP))}{\#CP}, \quad (10)$$

here, a negative MSD indicates that the changepoint detection algorithm tends to predict the changepoint window earlier than the true changepoint window, while a positive MSD indicates that the algorithm tends to predict the changepoint window later than the true changepoint window. To calculate the MSD of each changepoint detection algorithm on different changepoint categories, we compute the average MSD, which is given by

$$Avg.MSD_g = \frac{\sum_{i=1}^{N_g} MSD_i}{N_g}, \quad (11)$$

where $Avg.MSD_g$ represents the average MSD for changepoint category g , MSD_i is the MSD value for the i th effect in changepoint category g , and N_g is the number of effects in changepoint category g . This allows us to assess each changepoint detection algorithm's tendency to detect changepoints early or late in both univariate and multivariate changepoints scenarios of effects.

2.6. Real-life data utilization

To validate the effectiveness of our proposed changepoint detection method in practical use, as well as to verify the properties we concluded for each changepoint detection algorithm from synthetic data, we utilized all three changepoint detection algorithms on the real-life data. Specifically, we analyzed the publicly available Apollo 13 voice loop data, which recorded the communication between the astronauts and Mission Control during the failed Apollo 13 mission. The mission was intended to land on the Moon, but an agitation of one of the oxygen tanks caused an explosion that damaged the wire insulation inside, leading to the discharge of both oxygen tanks. This left the astronauts without systems to generate electricity and oxygen, forcing them to contact Mission Control for assistance and ultimately leading to the mission's cancellation. For our analysis, we selected a subset of the voice loop data covering the period from one hour before the emergency until Apollo 13 was safely back on a trajectory towards Earth. Our focus was on the relevant communication data from the mission timeline between 54:46:28 and 62:06:53 (hh:mm:ss).

In the Apollo 13 voice loop data, a pivotal moment occurred when an astronaut stated, "I believe we've had a problem here," at 55:55:21. This moment marks the start of the emergency, and we have selected it as the potential location of the changepoint in our study. We hypothesize that the old interaction patterns were disrupted after this point, making it

an ideal location to test the effectiveness of our proposed changepoint detection approach on a real social network.

To apply our proposed changepoint detection approach (see Algorithm 1), we selected several effects that shape the network. These effects include the “Inertia effect,” which refers to the tendency for actors to repeatedly interact with each other, the “Outdegree of the sender effect,” which refers to the tendency for actors to send events if they have sent more past events, the “Indegree of the receiver effect,” which refers to the tendency for actors to receive events if they have received more past events, and the “Total degree of the sender effect,” which refers to the tendency for actors to send events if they have sent and received more past events. We also selected three other effects that capture specific patterns of interaction: the “AB-BA pshift,” which refers to the tendency for immediate reciprocation, where the next sender is the current receiver and the next receiver is the current sender, the “AB-XA pshift,” which refers to a tendency for turn usurping, where the next sender is not in the current event and the next receiver is the current sender, and the “AB-BY pshift,” which refers to a tendency for turn receiving, where the next sender is the current receiver and the next receiver is not in the current event.

Considering the findings of Meijerink-Bosman’s 2022 research [29], we decide to fit the MW-REM with a 1000-second window length and $\frac{2}{3}$ overlap, as it provided a good insight into the dynamic of the social network. We then apply this window setting to the Apollo 13 voice loop data, extract the parameters of each effect along the windows, and feed them to the three changepoint detection algorithms employed in our study.

3. Results & Discussion

3.1. Synthetic REH datasets analysis

In this section, we present the results of the changepoint detection analysis conducted on the synthetic REH data. We summarize the performance of three changepoint detection algorithms under zero, one, and two changepoint conditions based on the following metrics: (1) the number of false positives, (2) recall, (3) precision, (4) average MSE, and (5) average MSD.

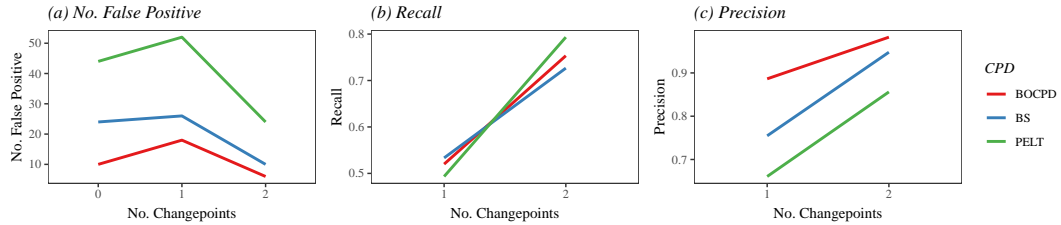


Figure 3. The plot displays the (a) number of false positives, (b) recall, and (c) precision of the three changepoint detection algorithms under no changepoint, one changepoint, and two changepoints conditions. The y-axes indicate the values of interest, the x-axes indicate the different changepoint categories. The red line represents BOCPD, the blue line represents BS, and the green line represents PELT.

The results of the simulation studies support the effectiveness of our proposed method, all three changepoint detection algorithms (BS, PELT, and BOCPD) demonstrated persuasive performance. In Figure 3 (a), we observe the overall performance of the algorithms in terms of false positives. The BOCPD algorithm outperforms the other algorithms in all scenarios, including no changepoint, one changepoint, and two changepoints scenarios. This suggests that the BOCPD algorithm has a better ability to correctly identify windows that are not changepoint windows in any situation. Conversely, the PELT algorithm is the most prone to incorrectly identify non-changepoint windows as changepoint windows.

Regarding the recall in Figure 3 (b), all three changepoint detection algorithms have similar recall values in the univariate changepoint condition, which are around 0.5. This indicates that if an effect has only one changepoint throughout the entire REH, the probability of detecting the window within a range of three windows from the true one is around 50%, using any of the three algorithms. In the case of the multivariate changepoint condition, all three algorithms show significant improvement, especially PELT, which almost reaches 80%.

In Figure 3 (c), we present the precision performance of the three algorithms. As with recall, all three algorithms show improved precision in the multivariate changepoint condition, with each achieving a precision of over 0.85. BOCPD consistently performs at a high level, regardless of the changepoint condition. This suggests that the changepoint windows it identified are highly likely to be the windows within three windows' range of the true changepoint window. The low precision and high recall of PELT, however, are a trade-off with its high false positives. PELT detects more changepoint windows, which increases the likelihood of covering more true positives in its predictions, but at the same time, the high false positives result in the lowest precision among the three algorithms. Details of the performance of each algorithm in terms of the three metrics are shown in Table 4.

	<i>BS</i>	<i>PELT</i>	<i>BOCPD</i>
Number of false negatives			
<i>No changepoint category</i>	24	44	10
<i>One changepoints category</i>	26	52	18
<i>Two changepoints category</i>	10	24	6
Recall			
<i>One changepoints category</i>	0.53	0.49	0.52
<i>Two changepoints category</i>	0.73	0.79	0.75
Precision			
<i>One changepoints category</i>	0.75	0.66	0.89
<i>Two changepoints category</i>	0.95	0.86	0.98

Table 4. False negatives, recall, and precision for each changepoint detection algorithm in various changepoint conditions.

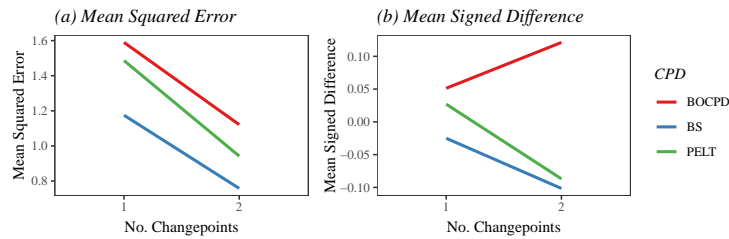


Figure 4. The plot shows the (a) average mean squared error, and (b) average mean signed difference under no changepoint, one changepoint, and two changepoints conditions. The y-axes indicate the values of interest, the x-axes indicate the different changepoint categories. The red line represents BOCPD, the blue line represents BS, and the green line represents PELT.

Figure 4 (a), the average MSE performance of the three algorithms is shown. The average MSE measures the average squared distance between the predicted and actual changepoint windows for different changepoint situations. A lower MSE value indicates that the predicted true positive changepoint windows are closer to the true changepoint windows. Among the three algorithms, BS performs the best in both univariate and multivariate changepoint situations, with the lowest average differences between its predicted and actual changepoint windows. However, the differences between the three algorithms are small overall.

Figure 4 (b) reveals the average MSD of each algorithm. The average MSD provides information about the average relative location of the predicted changepoint window with the true changepoint window in different changepoint situations for each algorithm. The BOCPD tends to detect the changepoint window after the true changepoint, whether in the univariate or multivariate changepoint context. In contrast, the BS tends to detect the window before the true changepoint, and PELT has mixed performance. The reason why BS and PELT tend to detect the changepoint window before the true changepoint is that they merge small intervals based on a cost function balancing data likelihood and model complexity[25]. This approach favors fewer, larger changepoints and may lead to detecting changepoints slightly before the true location¹⁵. Additionally, it is possible that the true changepoint window is located in a region of the time series where the effect parameters are already changing but have not yet fully transitioned to the new regime. Details of the performance of each algorithm in terms of the average MSE and average MSD metrics are shown in Table 5.

	<i>BS</i>	<i>PELT</i>	<i>BOCPD</i>
Average MSE			
<i>One changepoints category</i>	1.18	1.49	1.59
<i>Two changepoints category</i>	0.76	0.94	1.12
Average MSD			
<i>One changepoints category</i>	-0.03	0.03	0.05
<i>Two changepoints category</i>	-0.1	-0.09	0.12

Table 5. Average MSE and average MSD for each changepoint detection algorithm in various changepoint conditions.

Overall, the BOCPD performs well in terms of precision; the changepoint windows identified by the BOCPD are highly likely to be the true changepoints or within a three-window range of the true changepoints. However, in general, it cannot detect as many true changepoints as PELT. PELT has good performance on recall; the true changepoints are likely to be covered by all the predicted changepoints from PELT. However, this comes at the cost of high false positives and low precision. The BS performs well in terms of average MSE; the location of its predicted changepoint windows is usually close to the true changepoint window if the predicted changepoint windows are true positive cases (i.e., within the three-window range of the true changepoint windows). However, the weakness of the BS is that it has the lowest recall among the three algorithms, which can cause it to miss detecting certain changepoint windows. Moreover, in practice, it is challenging to determine whether a detected changepoint window is a true positive or not. Therefore, instead of relying solely on the BS, we suggest using it in conjunction with other algorithms. By comparing the nearby detected changepoint windows between the BS and other algorithms, we can identify the true positive changepoint windows and choose the window detected by the BS as it has good performance in terms of average MSE. Below shows the suitable application scenarios of each algorithm:

- BS: suitable when accurately identifying the location of the changepoint window is essential. Note that this algorithm needs to be used in conjunction with other algorithms as an auxiliary tool.
- PELT: appropriate when it is critical to identify all potential changepoint windows, even if it means a higher likelihood of false positives.
- BOCPD: recommended when proper identification of changepoint windows is crucial and minimizing the number of incorrect detections is a priority, even if it means potentially missing some true changepoint windows.

On the other hand, it is worth noting that all three algorithms perform better on multivariate changepoint conditions (i.e., scenarios with more than one changepoint) in all the metrics we employed. This is because detecting multiple changepoints can provide more information about

the underlying structure of the data. Conversely, detecting one or fewer changepoints may not provide enough information to distinguish between random fluctuations and actual changes[26].

3.2. Real-life data analysis

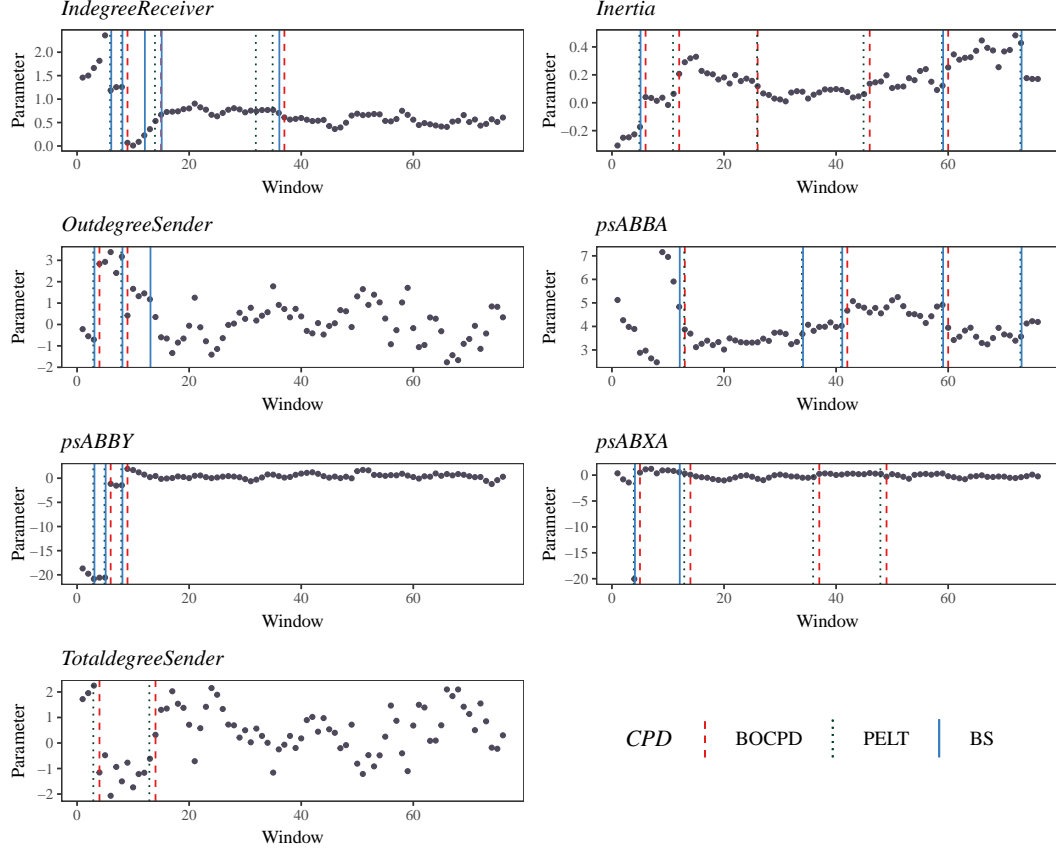


Figure 5. The plot shows the number of false positives, true positive rate, and positive predictive value of the three changepoint detection algorithms for effects without changepoint, effects with one changepoint, and effects with two changepoints.

The emergency report of Apollo 13 occurred at $t = 55:55:21$ (hh:mm:ss). The MW-REM was applied with a 1000-second window length and $\frac{2}{3}$ overlap, resulting in a total of 76 windows. The actual changepoint window for our study corresponds to the window from 9 to 11, covering the time slot from 55:38:53 to 56:06:40.

We inputted the effect parameters obtained from the MW-REM windows into three different changepoint detection algorithms: BOCPD, BS, and PELT. The resulting changepoint detections are displayed in Figure 5. It is worth noting that the emergency happened between windows 9 and 11, so we anticipated that the algorithms would detect the emergency during or around that timeframe.

For the indegree of the receiver effect, BOCPD detected changepoints at windows 9, 15, and 37. BS detected them at windows 6, 8, 12, 15, and 36. PELT detected them at windows 6, 8, 14, 32, and 35. BOCPD successfully captured the changepoint window at 9 and also detected window 15, which occurred four windows later than the actual changepoint window, BS and PELT did not capture any exact window that contained the emergency. BS detected

a window that occurred one window before it, and a window that occurred four windows later than it. PELT detected a window that occurred one window ahead of it, and another that was three windows later than it. As we mentioned in the previous section, BS and PELT tend to detect the changepoint window before the true changepoint due to their cost function [25], which favors fewer, larger changepoints and may lead to detecting changepoints slightly before the true location. Additionally, our study is a retrieval study that analyzed the complete data, not incomplete and updated data over time, which may have caused the algorithms to detect the changepoint window slightly earlier.

On the other hand, the fact that the algorithms detected the changepoint slightly after the window that contained the emergency can also be justified. The MW-REM focuses on the dynamics of communication between the actors, in our case, the astronauts and Mission Control. It's possible that their communications were not as frequent immediately after the emergency happened, but instead increased after some time had passed.

From the indegree of the receiver effect in Figure 5, we can also see a significant decrease in the parameter at the window the emergency happened, followed by an increase until window 15 when it stabilized. This indicates that after the issue was reported, the phenomenon of the actors to keep receiving events if they have received more past events decreased, then had a slight rebound. This is consistent with our expectations, as when the issue was reported, the old receivers changed, and the astronauts had more interactions with Mission Control but not between themselves. The staff at Mission Control became the new receivers, and this explains why the parameters rebounded afterward.

For the inertia effect, BOCPD detected changepoints at windows 6, 12, 26, 46, and 60. BS detected changepoints at windows 5, 59, and 73, while PELT detected changepoints at windows 5, 11, 26, 45, 59, and 73. BOCPD detected the changepoint one window after the window containing the emergency report, which is very close to the window where the emergency occurred. However, BS did not detect any relevant windows near the changepoint window. On the other hand, PELT recognized the changepoint at window 11, which is exactly the window where the emergency happened.

The inertia effect in Figure 5 also shows an increase in the inertia parameter after the emergency, which was successfully captured by BOCPD and PELT. The rise in the inertia effect could be because the emergency caused certain astronauts and staff at Mission Control to exchange more reports and responses to tackle the issue during that time period, resulting in a higher inertia effect. However, the effect later decreased slightly.

Regarding the outdegree of the sender effect, BOCPD detected changepoints at windows 4 and 9, BS detected them at windows 3, 8, and 13. PELT detected the changepoint at windows 3 and 8. Once again, BOCPD captured the window in which the emergency occurred, while BS and PELT detected the changepoint one window before it actually happened. The panel of the outdegree of the sender effect in Figure 5 reveals that the parameter of the outdegree of the sender effect had a significant reduction at window 9, the window of the emergency report, and then had a slight increase. This reflects that after the emergency occurred, the phenomenon of the actors who sent more past events would send more in the future was broken. To fix the issue, the astronauts and the staff at Mission Control who were not active senders before had to join the communications more. Therefore, the parameter dropped at window 9, and then they became more active actors than before, leading to a slight rise in the parameter of the outdegree of the sender effect after window 9.

For the AB-BA pshift effect, BOCPD detected changepoints at windows 13, 42, and 60, BS detected them at windows 12, 34, 41, 59, and 73, PELT detected them at windows 13, 34, 41, 59, and 73. Regarding the AB-BY pshift effect, BOCPD detected changepoints at windows 6 and 9, BS detected them at windows 3, 5, and 8, and PELT detected them at windows 3, 5, and 8. For the AB-XA pshift effect, BOCPD detected changepoints at windows 5, 14, 37, and 49, BS detected them at windows 4 and 12, and PELT detected them at windows 4, 13, 36, and 48. Except for BOCPD's performance on the AB-BY pshift effect, none of the three algorithms detected the window that contained the emergency report for these three effects. However,

they all detected the window very close to the emergency report window, within a range of three windows from when the emergency occurred. For the AB-BY pshift effect, BS and PELT once again detected the changepoint one window ahead of the emergency report window.

From the AB-BA pshift effect panel shown in Figure 5, it can be observed that after the window containing the emergency, the parameter for the AB-BA pshift effect significantly declined. This suggests a reduction in the immediate reciprocation, where the next sender becomes the current receiver and the next receiver becomes the current sender. Similar to the AB-BA pshift effect, after the window containing the emergency, the parameter for the AB-XA pshift effect also dropped, but not as strongly. This indicates a slight reduction in the pattern where the next sender is not in the current event and the next receiver is the current sender. However, in contrast to the AB-BA pshift effect and the AB-XA pshift effect, the AB-BY pshift effect panel shows an increase in the parameter after the window containing the emergency, suggesting an ascending trend of the phenomenon where the current receiver becomes the next sender and the next receiver is not in the current event. Overall, combining the parameter fluctuations around the changepoints of the three effects, it becomes apparent that the sender was not receiving an immediate response from either the event’s receiver or other actors as frequently after the emergency occurred. Instead, the receiver was more likely to become the sender of the following event. This suggests that after the emergency, it may be more common for the astronauts to send messages to the staff at Mission Control, who would then transmit or discuss the plan or solutions with other staff at Mission Control.

Regarding the total degree of the sender effect, BOCPD detected the changepoints at window 4 and 14, PELT detected the changepoints at window 3 and 13. In contrast, BS did not detect any changepoint throughout the entire REH of the total degree of the sender effect. Although none of the algorithms detected the exact window involving the emergency, BOCPD and PELT identified nearby windows, specifically three windows after and two windows later than the emergency report window, respectively. As shown in the Figure 5, the total degree of the sender effect reveals a sudden increase in the parameter following the window that contained the emergency report. This suggests that during the Apollo 13 mission, both the astronauts and Mission Control became more active in sending and receiving messages after the emergency. It is possible that the staff who had more prior communication with the astronauts provided more instructions to help address the issues. The observed phenomenon, where actors send more events if they have sent and received more past events, seems to have played a significant role in the communication dynamics after the emergency.

Overall, we validated our proposed approach for using the MW-REM structure to detect changepoints in social networks with real-life data from the Apollo 13 mission. Specifically, we analyzed the voice-loop data for the emergency report that occurred at $t = 55:55:21$ (hh:mm:ss). Using a 1000-second and $\frac{2}{3}$ overlap MW-REM, we found that windows 9 to 11 out of a total of 76 windows covered the emergency report. We hypothesized that by inputting each effect’s parameters into the changepoint detection algorithm, we could identify the window containing the emergency report. Our analysis confirmed this hypothesis, as all three changepoint detection algorithms we used performed well. They identified the changepoints either at or near the window that contained the emergency report for almost all of the effects.

The Apollo 13 analysis also confirmed the characteristic conclusions of each changepoint detection algorithm as observed in the synthetic data analysis. BOCPD had a high PPV and detected only a small number of changepoint windows, which were mostly the emergency report windows. PELT had the best TPR performance among the three algorithms, but detected a larger number of changepoint windows, including those around or at the emergency report windows for all effects. BS had the worst TPR performance and did not detect any changepoint windows around the emergency for the inertia effect and total degree of the sender effect, while BOCPD and PELT did. On the other hand, both BS and PELT identified the window one step before the emergency report window as a changepoint window for the indegree of the receiver effect, the outdegree of the sender effect, and the AB-BY pshift effect. This is consistent with their performance from the synthetic data analysis, where they possess the negative values of the MSD.

4. Conclusion

This paper proposes an approach based on the MW-REM structure to detect changepoints in social networks. Additionally, we compare the performance of three widely used changepoint detection algorithms, BOCPD, BS, and PELT, under the MW-REM structure. The proposed approach first identifies the key effects that influence social interactions in the network, and then selects an appropriate window length and overlap to segment the event history of the network. The MW-REM is then fit to the event history, and effect parameters are extracted for each fitting window of each effect. Finally, the extracted effect parameters are inputted into the changepoint detection algorithms to detect the changepoints in the network of each effect. With this approach, we can identify significant changes in the network over time and gain insights into the underlying dynamics of the network.

The validation of our approach using synthetic data and real-life data from the Apollo 13 mission has demonstrated its effectiveness in detecting changepoints in social networks. Both the synthetic data analysis and the Apollo 13 mission analysis yielded similar conclusions: the proposed approach is capable of detecting most of the changepoints in the multivariate changepoint contexts of the social network, as all three algorithms had a high true positive rate (TPR) in the synthetic data and spotted almost all the effects' windows containing or nearing the emergency report in the Apollo 13 analysis. BOCPD performed well in precision (PPV), tending to give fewer changepoint windows but which were mostly true changepoint windows. PELT had a good TPR but a higher number of false negative cases. PELT and BS sometimes detected the changepoint window a bit earlier than the actual changepoint window in the retrieval study due to their interval merging property, where they recognize a changepoint in a certain interval.

Overall, our approach provides a useful tool for detecting changepoints in social networks, which can be applied in various fields, such as social science, economics, and epidemiology, to identify critical changes in complex systems. Further research can explore the potential of our approach in identifying the exact time point of changepoints in the event history. As our proposed method identifies the changepoint by window periods, it infers the changepoint is within a certain time period, but not the specific time point in the event history. Shafiee Kamalabad, and Leenders et al.[23] suggested a Bayes factor changepoint detection method under the REM, which grids the time points of the event history and uses the support of two hypotheses from the REH to prove the existence of changepoints. By combining our method with the Bayes factor changepoint detection method, we can first identify the time period that the changepoint belongs to using our approach. Then, we can conduct the Bayes factor changepoint detection method on that period, which may provide us with the specific time point of the changepoint. This may be a promising area for further research. Additionally, future research can investigate the suitability of our approach in identifying changes in different types of networks and compare the performance of different changepoint detection algorithms under various network structures.

References

- [1] R. P. Adams and D. J. C. MacKay. Bayesian Online Changepoint Detection.
- [2] R. Alami, O. Maillard, and R. Feraud. Restarted Bayesian Online Change-point Detector achieves Optimal Detection Delay. In *Proceedings of the 37th International Conference on Machine Learning*, pages 211–221. PMLR.
- [3] S. Aminikhanghahi and D. J. Cook. A Survey of Methods for Time Series Change Point Detection. 51(2):339–367.
- [4] S. Aminikhanghahi and D. J. Cook. A Survey of Methods for Time Series Change Point Detection. 51(2):339–367.
- [5] I. E. Auger and C. E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. 51(1):39–54.
- [6] I. E. Auger and C. E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. 51(1):39–54.
- [7] C. T. Butts. A Relational Event Framework for Social Action. 38(1):155–200.
- [8] C. T. Butts and C. S. Marcum. A Relational Event Approach to Modeling Behavioral Dynamics. In A. Pilny and M. S. Poole, editors, *Group Processes*, pages 51–92. Springer International Publishing.
- [9] M. Chapman. A Meta-Analysis of Metrics for Change Point Detection Algorithms.
- [10] Y. Chen, T. Wang, and R. J. Samworth. Inference in high-dimensional online changepoint detection.
- [11] H. Cho and P. Fryzlewicz. Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. 77(2):475–507.
- [12] H. Dehling, R. Fried, and M. Wendler. A robust method for shift detection in time series. 107(3):647–660.
- [13] p. d. u. family=Burg, given=Gerrit J. J. and C. K. I. Williams. An Evaluation of Change Point Detection Algorithms.
- [14] p. d. u. family=Burg, given=Gerrit J. J. and C. K. I. Williams. An Evaluation of Change Point Detection Algorithms.
- [15] P. Fearnhead and G. Rigai. Changepoint Detection in the Presence of Outliers. 114(525):169–183.
- [16] P. Fryzlewicz. Wild binary segmentation for multiple change-point detection. 42(6).
- [17] X. Gu, J. Mulder, and H. Hoijtink. Approximated adjusted fractional Bayes factors: A general method for testing informative hypotheses. 71(2):229–261.
- [18] H. Hoijtink, X. Gu, and J. Mulder. Bayesian evaluation of informative hypotheses for multiple populations. 72(2):219–243.
- [19] N. Itoh. Change-Point Detection of Climate Time Series by Nonparametric Method.
- [20] N. Itoh and J. Kurths. Change-Point Detection of Climate Time Series by Nonparametric Method.
- [21] B. Jackson, J. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumoussis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and T. T. Tsai. An algorithm for optimal partitioning of data on an interval. 12(2):105–108.
- [22] D. Jarušková. Some Problems with Application of Change-Point Detection Methods to Environmental Data. 8(5):469–483.
- [23] M. S. Kamalabad, R. Leenders, and J. Mulder. What’s the Point of Change? Change Point Detection in Relational Event Models. page 21.
- [24] L. Kendrick, K. Musial, and B. Gabrys. Change point detection in social networks—Critical review with experiments. 29:1–13.
- [25] R. Killick, P. Fearnhead, and I. A. Eckley. Optimal Detection of Changepoints With a Linear Computational Cost. 107(500):1590–1598.
- [26] Y. Li, G. Lin, T. Lau, and R. Zeng. A Review of Changepoint Detection Models.
- [27] L. Ma, A. J. Grant, and G. Sofronov. Multiple change point detection and validation in autoregressive time series data. 61(4):1507–1528.
- [28] M. Meijerink-Bosman, M. Back, K. Geukes, R. Leenders, and J. Mulder. Discovering trends of social interaction behavior over time: An introduction to relational event modeling: Trends of social interaction.
- [29] M. Meijerink-Bosman, R. Leenders, and J. Mulder. Dynamic relational event modeling: Testing, exploring, and applying. 17(8):e0272309.

- [30] J. Mulder and R. T. Leenders. Modeling the evolution of interaction behavior in social networks: A dynamic relational event approach for real-time analysis. 119:73–85.
- [31] A. Rauhameri, K. Salminen, J. Rantala, T. Salpavaara, J. Verho, V. Surakka, J. Lekkala, A. Vehkaoja, and P. Müller. A comparison of online methods for change point detection in ion-mobility spectrometry data. 14:100151.
- [32] J. Reeves, J. Chen, X. L. Wang, R. Lund, and Q. Q. Lu. A Review and Comparison of Changepoint Detection Techniques for Climate Data. 46(6):900–915.
- [33] M. Shafiee Kamalabad and M. Grzegorzcyk. Improving nonhomogeneous dynamic Bayesian networks with sequentially coupled parameters. 72(3):281–305.
- [34] M. Shafiee Kamalabad and M. Grzegorzcyk. A new Bayesian piecewise linear regression model for dynamic network reconstruction. 22(2):196.
- [35] M. Shafiee Kamalabad and M. Grzegorzcyk. Non-homogeneous dynamic Bayesian networks with edge-wise sequentially coupled parameters. 36(4):1198–1207.
- [36] M. Shafiee Kamalabad, R. Leenders, and J. Mulder. What is the Point of Change? Change Point Detection in Relational Event Models. 74:166–181.
- [37] S. Sharma, D. A. Swayne, and C. Obimbo. Trend analysis and change point techniques: A survey. 1(3):123–130.
- [38] C. Truong, L. Oudre, and N. Vayatis. Selective review of offline change point detection methods. 167:107299.
- [39] G. D. Wambui, G. A. Waititu, and A. Wanjoya. The Power of the Pruned Exact Linear Time(PELT) Test in Multiple Changepoint Detection. 4(6):581.